



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
24.03.1999 Bulletin 1999/12

(51) Int Cl.⁶: **G05B 19/418**, G05B 19/042,
H04J 3/06, G06F 1/14

(21) Application number: **98307645.6**

(22) Date of filing: **21.09.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(72) Inventor: **Edison, John C.**
Palo Alto, California 94303 (US)

(74) Representative: **Powell, Stephen David**
WILLIAMS, POWELL & ASSOCIATES,
4 St Paul's Churchyard
London EC4M 8AY (GB)

(30) Priority: **22.09.1997 US 934792**

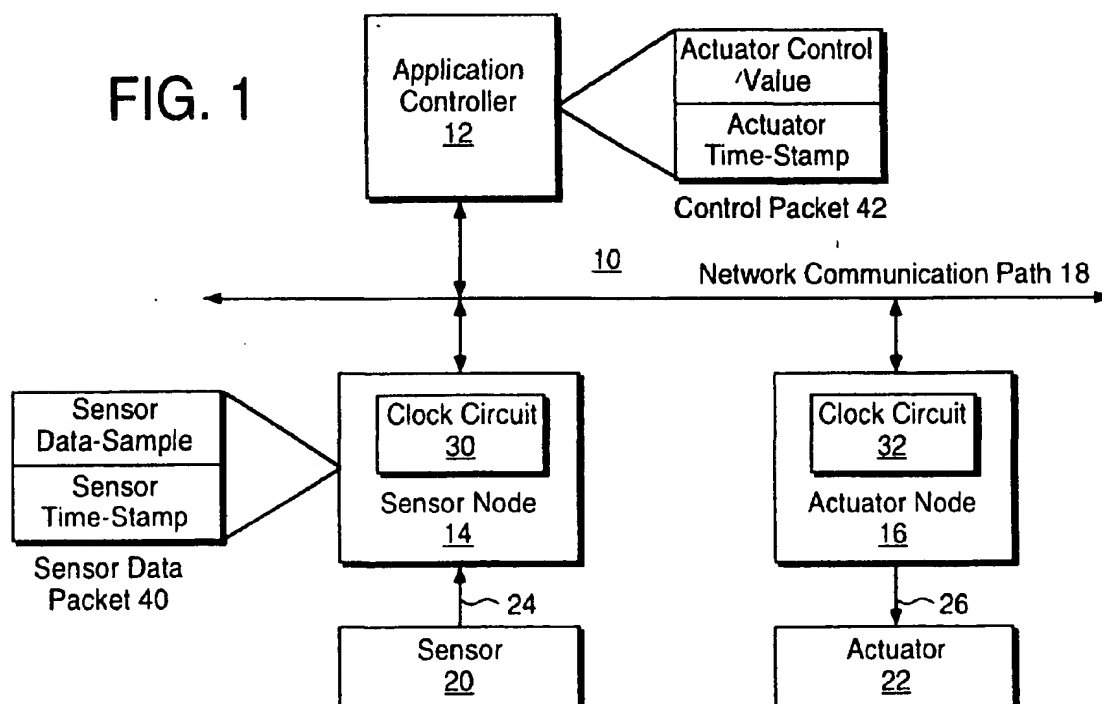
(71) Applicant: **Hewlett-Packard Company**
Palo Alto, California 94304 (US)

(54) **Control system with nodes**

(57) A control system (10), which provides accurate timing information for sensor data samples and accurate timing for the application of control values to actuators (22) in the face of nondeterministic communication, em-

plays circuitry for accurately associating sensor data samples and sensor time-stamps and circuitry for triggering the application of the control values to an actuator (22).

FIG. 1



Description

[0001] The present invention pertains to the field of control systems. More particularly, this invention relates to real-time control systems that employ non-deterministic communication.

[0002] Real-time control systems commonly include sensors and actuators and application controllers which are arranged to provide control of devices including devices used in industrial processes. Such sensors may include temperature sensors, pressure sensors, tachometers, etc. Such actuators may include, valves, motors, heaters etc. Application controllers in such a control system may be implemented with programmable logic controllers (PLCs) or computer systems including personal computer systems.

[0003] A real-time control system in its simplest form includes an application controller, at least one sensor, and at least one actuator. Typically, the application controller implements a control algorithm which is adapted to maintain a set point for a particular device being controlled. Typically, the application controller obtains sensor data samples from the sensor and uses the sensor data samples as inputs to a control algorithm which computes a control value to be applied to the actuator. The application controller then usually writes the control value to the actuator. Typically, the application controller continually obtains sensor data samples and continually provides control values to the actuator in order to maintain a desired set point. Such an arrangement may be referred to as a closed-loop control system.

[0004] For example, a simple real-time control system for a motor may include a tachometer (sensor) that measures the rotational speed of the motor and an amplifier circuit (actuator) that increases or decreases the rotational speed of the motor in response to a control value applied to the amplifier circuit. An application controller periodically obtains rotational speed samples from the tachometer and periodically writes control values to the amplifier circuit in order to maintain the desired set point for the speed of the motor.

[0005] A typical control algorithm for such a real-time control system typically uses timing information together with sensor data samples to compute control values. In addition, such a control algorithm usually associates timing information with each control value. One such type of control algorithm is referred to as a proportional integral derivative (PID) algorithm. A PID algorithm usually takes as input a time value associated with each sensor data sample. The time values enable the PID algorithm to take into account the rate of change of the sensor data samples when computing a new control value. In addition, the PID algorithm usually assigns a time value to each new control value which indicates a time at which the new control value is to be applied to the actuator.

[0006] As a consequence, a real-time control system must usually provide accurate timing information for

each sensor data sample in order to render accurate computation of control values. In addition, a real-time control system must typically apply each control value to the actuator with accurate timing in order to provide accurate control of the desired set point. Unfortunately, prior control systems commonly suffer from inaccuracies in the relationship between the sensor data samples and their associated timing information. In addition, inaccuracies commonly exist as to the time that a control value is applied to an actuator in prior systems. In general, such uncertainties are caused by the non-deterministic nature of communication that is commonly employed in prior real-time control systems.

[0007] For example, the sensors, actuators, and controllers in prior real-time control systems are commonly interconnected via a communication network. Such a communication network may be implemented with a packet-based communication protocol that includes collision handling. Ethernet is an example of a packet-based network with collision handling. Another example is LonTalk which is a field-level control bus specialized for the process control environment. Such a communication network may also be implemented with time division multiple access (TDMA) or token ring protocols to name a few.

[0008] An application controller connected to such a communication network typically obtains sensor data samples using messages transferred over the communication network. Typically, the application controller includes a real-time clock which is used to assign a time value to each sensor data sample received over the communication network.

[0009] Typically, a delay exists between the time a sensor data sample is generated and the time an application controller obtains a time value for that sensor data sample from its real-time clock. Such delay usually includes the delay associated with message transfer over the communication path. Such delay also typically includes the delay associated with the application controller obtaining a time value from its real-time clock.

[0010] In addition, a delay usually exists between the time the time that an application controller generates a control value and the time that the control value is applied to an actuator. Such delay usually includes the delay associated with message transfer over the communication path to the actuator as well as the delay in the actuator before the control value is applied.

[0011] Such delays, both in sensor data timing and in the application of control values, are usually not constant. Instead, such delays typically vary in a non-deterministic or random manner. Such variation in delay may be referred to as jitter. Jitter may be caused by the variation of traffic on the communication network. In addition, collisions that occur on the communication network can introduce substantial amounts of jitter. Other sources of jitter may include variation in the time taken by an operating system, under which the application controller usually runs, to render a real-time clock value in the face

of varying amounts of operating system activity.

[0012] Unfortunately, such jitter usually creates inaccuracies in associating time values to sensor data samples. Such jitter also typically creates inaccuracies in the timing of the application of the control values to actuators. Such inaccuracies typically decreases the accuracy and efficiency of such prior control systems.

[0013] A control system is disclosed that provides accurate timing information for sensor data samples and accurate timing for the application of control values to actuators in the face of non-deterministic communication. The control system includes a sensor node with circuitry that generates a sensor time-stamp substantially contemporaneously with a time at which it obtains a sensor data sample. The sensor time-stamp enables accurate computation of an actuator control value and an actuator time-stamp. The control system includes an actuator node having circuitry for triggering the application of the actuator control value to an actuator using the actuator time-stamp.

[0014] Other features and advantages of the present invention will be apparent from the detailed description that follows.

[0015] The present invention is described with respect to particular exemplary embodiments thereof and reference is accordingly made to the drawings in which:

Figure 1 illustrates one embodiment of a real-time control system that provides accurate timing information for sensor data samples and accurate timing for the application of control values to actuators in the face of non-deterministic communication;

Figure 2 illustrates one embodiment of mechanisms in a sensor node for accurately time-stamping sensor data samples;

Figure 3 illustrates one embodiment of a mechanism in an actuator node for accurately applying actuator control values to the actuator;

Figure 4 shows a process control network which implements methods for handling event storms.

[0016] **Figure 1** illustrates one embodiment of a real-time control system 10 that provides accurate timing information for sensor data samples and accurate timing for the application of control values to actuators in the face of non-deterministic communication. The real-time control system 10 includes an application controller 12, a sensor node 14 and an actuator node 16. The sensor node 14 includes circuitry that enables accurate time-stamping of sensor data samples of a sensor 20. The actuator node 16 includes circuitry that enables the accurate application of control values to an actuator 22.

[0017] The network communication path 18 may be implemented with one of a variety of communication mechanisms. In one embodiment, the network commu-

nication path 18 is an Ethernet communication network. In another embodiment, the network communication path 18 is a LonTalk field-level control bus which is specialized for the process control environment. In other embodiments, the network communication path 18 may be implemented with time division multiple access (TDMA) or token ring protocols to name only a few possibilities.

[0018] The sensor 20 represents any sensor that may be employed in a real-time control system. Such sensors include for example devices such as temperature sensors, pressure sensors, and tachometers. The actuator 20 represents any actuator that may be employed in a real-time control system. Such actuators include for example devices such valves, motors, heaters, and amplifiers.

[0019] The sensor node 14 interfaces the sensor 20 to the network communication path 18. The sensor node 14 includes the communication hardware and software necessary for communication via the network communication path 18. The sensor node 14 also includes a clock circuit 30 that enables accurate time-stamping of sensor data samples obtained from a sensor 20.

[0020] The actuator node 16 interfaces the actuator 22 to the network communication path 18. The actuator node includes the communication hardware and software necessary for communication via the network communication path 18. The actuator node 16 also includes a clock circuit 32 that enables the accurate application of control values to an actuator 22. In one embodiment, the clock circuits 30 and 32 are synchronized by transferring time packets via the network communication path 18 and providing the sensor node 14 and the actuator node 16 with time packet detectors in a manner described in U.S. Patent No. 5,566,180.

[0021] The application controller 12 implements a process control algorithm for closed loop control of a device (not shown). The application controller 12 senses the performance of the device using the sensor 20 and controls the performance of the device using the actuator 22. For example, the device may be a motor and the actuator 22 may be an amplifier that controls the speed of the motor and the sensor 20 a tachometer that measures the speed of the motor. The application controller 12 includes the communication hardware and software, i.e. protocol stack, necessary for communication via the network communication path 18. The application controller 12 may be implemented as programmable logic controller (PLC) or as a computer system.

[0022] The sensor node 14 generates a sensor data sample by sampling a sensor data signal 24 from the sensor 20. At the same time the sensor node 14 obtains a sensor time-stamp from the clock circuit 30. The sensor time-stamp is obtained from the clock circuit 30 substantially contemporaneously with the sampling of the sensor data signal 24 in order to provide an accurate sensor time-stamp. In one embodiment, the sensor node 14 samples the sensor data signal 24 in response

to a request message for sensor data that is received via the network communication path 18. Such an embodiment may be referred to as a pull system in that an external node pulls data from the sensor node 14. The request for sensor data may originate with the application controller 12 or with the actuator node 16.

[0023] The sensor node 14 generates a sensor data packet 40. The sensor data packet 40 includes the sensor data sample obtained from the sensor 20 and the sensor time-stamp obtained from the clock circuit 30. The sensor node 14 transfers the sensor data packet 40 via the network communication path 18. The sensor data packet 40 in a pull system is transferred to the node on the network communication path 18 that issued the request for sensor data which may be either the application controller 12 or the actuator node 16.

[0024] In another embodiment, the sensor node 14 samples the sensor data signal 24 and obtains a corresponding sensor time-stamp without any request messages. This embodiment may be referred to as a push system. In a push system, the sensor node 14 is pre-programmed to periodically obtain sensor data samples and sensor time stamps. The sensor node 14 periodically transfers the sensor data packet 40 containing newly obtained sensor data samples and sensor time-stamps over the network communication path 18 to either the application controller 12 or the actuator node 16.

[0025] In the following discussion it is assumed that the application controller 12 implements the process control algorithm. The application controller 12 receives the sensor data packet 40 via the communication path 18 and uses the sensor data sample and sensor time-stamp contained therein as inputs to its process control algorithm. The process control algorithm generates an actuator control value and a corresponding actuator time-stamp which indicates a future time at which the actuator control value is to be applied to the actuator 22. The application controller 12 assembles actuator control value and actuator time-stamp into a control packet 42. The application controller 12 then transfers the control packet 42 to the actuator node 16 via the network communication path 18.

[0026] The actuator node 16 receives the control packet 42 via the network communication path 18 and uses the actuator time-stamp contained therein as a triggering event for applying the actuator control value to the actuator 22 as an actuator signal 26. The actuator node 16 synchronizes the application of the actuator control value to the actuator time-stamp using the clock circuit 32. The clock circuit 32 is synchronized to the clock circuit 30 and the application controller 12 derives the actuator time-stamp from the sensor time-stamp which was generated by the clock circuit 30.

[0027] The process control system 10 shown depicts an embodiment in which the application controller 12 obtains sensor data samples and time-stamps from the sensor node 14 via the network communication path 18, computes actuator control values and actuator time-

stamps, and then transfers the actuator control values and time-stamps to the actuator node 16 via the communication path 18. In another embodiment, the process control algorithm may be implemented in the actuator node 16 which includes a processor and other software execution resources. In such an embodiment, the actuator node 16 obtains sensor data samples and time-stamps from the sensor node 14 via the communication path 18, computes actuator control values and time-stamps, and applies the actuator control values to the actuator 22.

[0028] In yet another embodiment, the process control algorithm for the device attached to the actuator 22 is implemented in the sensor node 14 which includes a processor and other software execution resources. In such an embodiment, the sensor node 14 obtains sensor data samples and time-stamps, computes actuator control values and time-stamps, and then transfers the actuator control values and time-stamps to the actuator node 16 via the network communication path 18.

[0029] Figure 2 illustrates mechanisms in the sensor node 14 for accurately time-stamping sensor data samples in one embodiment. The sensor node 14 includes a processor 52 that obtains a sensor data sample 62 from an analog-to-digital converter 54 and a sensor time-stamp 64 from the clock circuit 30 and then assembles the sensor data packet 40. The sensor node 14 also includes a protocol stack 50 which includes the software and hardware elements that enable communication via the network communication path 18.

[0030] The analog-to-digital converter 54 is coupled to receive the analog sensor data signal 24 from the sensor 20. The analog-to-digital converter 54 generates the sensor data sample 62 by digitizing the sensor data signal 24 in response to a strobe signal 60 generated by the processor 52. The analog-to-digital converter 54 holds the sensor data sample 62 for subsequent reading by the processor 52. The strobe signal 60 also causes a latch 56 to sample and hold a clock value 66 from the clock circuit 30. The strobe signal 60 causes the latch 56 to sample the clock value 66 substantially contemporaneously with the sampling of the sensor data signal 24 by the analog-to-digital converter 54.

[0031] In one embodiment, the processor 52 issues the strobe signal 60 to sample the clock value 66 and the sensor data signal 24 in response to a request message received via the network communication path 18. In another embodiment, the processor 52 periodically issues the strobe signal 60 to sample the clock value 66 and the sensor data signal 24. For example, the processor 52 may read or "poll" the clock value 66 to determine when another set of samples of the clock value 66 and the sensor data signal 24 should be obtained.

[0032] The processor 52 reads the sensor data sample 62 held by the analog-to-digital converter 54 and writes it into the sensor data packet 40. In addition, the processor 52 reads the sensor time-stamp 64 held by the latch 56 and writes it into the sensor data packet 40.

The processor 52 then transfers the sensor data packet 40 over the network communication path 18 using the protocol stack 50. The sensor data packet 40 in pull system is transferred to the requesting node, for example the application controller 12. In a push system, the processor 52 transfers the sensor data packet 40 to either the application controller 12 or the actuator node 16 whichever is predetermined to be the target.

[0033] Figure 3 illustrates one embodiment of a mechanism in the actuator node 16 for accurately applying actuator control values to the actuator 22. The actuator node 16 includes a protocol stack 70 which includes the software and hardware elements that enable communication via the network communication path 18. The actuator node 16 also includes a processor 72 that receives the control packet 42 via the network communication path 18 using the protocol stack 70.

[0034] The processor 72 writes the actuator control value from the control packet 42 to a digital-to-analog converter 74 as a control value 80. The digital-to-analog converter 74 holds the control value 80 until a strobe signal 88 is asserted by a comparator 78. In addition, the processor 72 writes the actuator time-stamp contained in the control packet 42 into a latch 76.

[0035] The comparator 78 compares a latched actuator time-stamp 82 with a free running clock value 86 generated by the clock circuit 32. The comparator 78 asserts the strobe signal 88 when the free running clock value 86 reaches the latched actuator time-stamp 82. The strobe signal 88 causes the digital-to-analog converter 74 to convert the control value 80 into an actuator control signal 26 for the actuator 22.

[0036] In another embodiment, the triggering function provided by the comparator 78 is instead performed in firmware, for example, with a compare instruction executed by the processor 72.

[0037] Figure 4 shows a process control network 100 which implements methods for handling event storms. The process control network 100 includes a sensor node 104 and an actuator node 106 that together implement a control loop. The sensor node 104 and the actuator node 106 each include a synchronized clock circuit and the hardware and software elements for communication via a network communication path 102. The sensor node 104 periodically transfers a sensor data packet containing a sensor data sample and a corresponding sensor time-stamp to the actuator node 106 via the network communication path 102.

[0038] In one embodiment, the sensor node 104 and the actuator node 106 provide a pull system in which the actuator node 106 periodically transfers a request messages to the sensor node 104 via the network communication path 102. The sensor node 104 responds to each request message by obtaining a sensor data sample and a corresponding sensor time-stamp and transferring them in a sensor data packet via the network communication path 102. In another embodiment, the sensor node 104 and the actuator node 106 provide a

push system in which the sensor node 104 periodically obtains a sensor data sample and a corresponding sensor time-stamp and transfers them in a sensor data packet via the network communication path 102.

[0039] The process control network 100 includes a set of monitor nodes 110-114. Each of the monitor nodes 110-114 performs a monitoring function to detect an unusual event or alarm in the process control network 100. These alarms may be, for example, over-temperature alarms or other events that must be communicated to other parts of the process control network 100.

[0040] A condition may occur in the process control network 100 which may be referred to as an event storm. An event storm may occur, for example, when a large number of the monitor nodes 110-114 detect an unusual event. The monitor nodes 110-114 that detect an unusual event transfer alarm messages via the network communication path 102 to notify other parts of the process control network 100 of the unusual events.

[0041] The large number of alarm messages transferred via the network communication path 102 during an event storm can consume most of the available bandwidth of the network communication path 102. These event storms can cause large numbers of collisions and the resulting collision arbitration mechanism on the network communication path 102 can slow overall throughput in the process control network 100. As a consequence, the messages required to maintain the control loop between the sensor node 104 and the actuator node 106 can become delayed or even lost or transferred out of order.

[0042] The sensor time-stamps contained in the sensor data packets transferred over the network communication path 102 enable the detection of delayed or missing or out of order sensor data packets. For example, assume the control loop between the sensor node 104 and the actuator node 106 includes the sampling of sensor data every 10 seconds. If so, then a sensor data packet should be transferred via the network communication path 102 approximately every 10 seconds and the sensor time-stamp contained in each sensor data packet should differ from a previous one by approximately 10 seconds. If two consecutive sensor data packets carried on the network communication path 102 have time-stamp values that differ by significantly more than 10 seconds then it is an indication that sensor data packets are being delayed or lost possibly due to an event storm.

[0043] Any one or more of the actuator node 106 and the monitor nodes 110-114 may use the sensor time-stamps in the sensor data packets transferred via the network communication path 102 to detect event storms. The actuator node 106 as the target already receives the sensor data packets and extracts the sensor time-stamps to compute actuator control values. In addition, the monitor nodes 110-114 can monitor or "snoop" the network communication path 102 for sensor data packets and record the sensor time-stamps to detect delayed or missing sensor data packets.

[0044] In one embodiment, if for example the monitor node 110 detects delayed or missing sensor data packets that may signal an event storm it stops transferring monitor messages on the communication path 102 for a predetermined time period. In another embodiment, the monitor node 110 transfers an emergency notification message via the communication path 102 if it detects a possible event storm. The emergency notification message causes the remaining monitor nodes 112-114 to stop transferring messages on the communication path 102 for a predetermined time period.

[0045] Another method for detecting possible event storms involves the detection of collisions on the network communication path 102 for embodiments where the network communication path 102 provides collision handling such as Ethernet. For example, the protocol stack, i.e. the communication hardware and software, in the sensor node 104 includes a mechanism for logging the occurrence of collisions on the network communication path 102. The processor in the sensor node 104 periodically queries the corresponding protocol stack to determine whether excessive numbers of collisions are occurring which could indicate an event storm.

[0046] In another embodiment, the protocol stack in the sensor node 104 includes a outbound message queue for buffering messages for transfer via the network communication path 102. The processor in the sensor node 104 periodically checks the outbound message queue. A nearly full outbound message queue could indicate that an event storm is underway.

[0047] In one embodiment, the sensor node 104 and the actuator node 106 run their control loop in a reduced performance mode if an event storm is detected. In one embodiment of a reduced state, the sensor node 104 provides sensor data samples every t_0/K seconds where K is a constant rather than the t_0 samples per second in a normal state without an event storm. The reduced state reduces the bandwidth utilization on the network communication path 102 required to maintain the control loop.

[0048] In another embodiment, the sensor node 104 and the actuator node 106 put the control loop into a "safe" state if an event storm is detected. The safe state may correspond to the last control value successfully received by the actuator node 106. Alternatively, the actuator node 106 may turn off the actuator once an event storm is detected.

[0049] The monitor nodes 110-114 may take a variety of actions once an event storm is detected. For example, the monitor nodes 110-114 can stop transferring messages on the network communication path 102, including alarm messages, for a period of time. The monitor nodes 110-114 may continue to obtain monitoring data and may include synchronized clock circuits that enable the monitor data to be accurately time-stamped. This would enable the monitor nodes 110-114 to transfer the buffered monitor data and time-stamps via the network communication path 102 once the event storms end and

allow a receiving node to reconstruct the monitor data with accurate timing information. Alternatively, the monitor nodes 110-114 can reduce that rate at which they transfer messages on the network communication path 102 for a period of time during an event storm.

[0050] The foregoing detailed description of the present invention is provided for the purposes of illustration and is not intended to be exhaustive or to limit the invention to the precise embodiment disclosed. Accordingly, the scope of the present invention is defined by the appended claims.

Claims

1. A control system comprising a sensor node (14) for interfacing a sensor (20) to a communication path (18), the sensor node (14) having a circuit for obtaining a sensor data sample and substantially contemporaneously generating a sensor time-stamp such that the sensor data sample and the sensor time-stamp enable an accurate computation of an actuator control value to be applied an actuator (22) in the control system.
2. The control system of claim 1, wherein the sensor data sample and the sensor time-stamp enable an accurate computation of an actuator time-stamp that indicates a time at which the actuator control value is to be applied to the actuator (22).
3. The control system of claim 2, further comprising an actuator node (16) for interfacing the actuator (22) to the communication path (18), the actuator node (16) having a circuit for triggering the application of the actuator control value to the actuator (22) using the actuator time-stamp.
4. The control system of claim 3, further comprising an application controller node (12) coupled to receive the sensor data sample and the sensor time-stamp via the communication path (18), the application controller node (12) generating the actuator control value and the actuator time-stamp in response to the sensor data sample and the sensor time-stamp.
5. The control system of claim 4, wherein the actuator node (16) receives the actuator control value and the actuator time-stamp via the communication path (18) from the application controller node (12).
6. The control system of claim 3, wherein the actuator node (16) includes a processor (72) that generates the actuator control value and the actuator time-stamp in response to the sensor data sample and the sensor time-stamp received via the communication path (18).

7. The control system of claim 1, wherein the sensor node (14) obtains the sensor data sample and generates the sensor time-stamp in response to a request message received via the communication path (18).
8. The control system of claim 1, wherein the sensor node (14) periodically obtains the sensor data sample and generates the sensor time-stamp without a request message being received via the communication path (18).
9. The control system of claim 1, wherein the sensor node (14) transfers a series of sensor data packets (40) via the communication path (18), each sensor data packet (40) containing a time-stamp such that the time-stamps carried on the communication path (18) enable the detection of an event storm in the process control system.
10. The control system of claim 9, further comprising a monitor node (110) that snoops the communication path (18) and examines the time-stamps to determine whether one or more of the sensor data packets (40) is missing or delayed.
11. The control system of claim 10, wherein the monitor node (110) transfers an emergency notification message via the communication path (18) if one or more of the sensor data packets (40) is missing or delayed.
12. The control system of claim 11, wherein the emergency notification message causes one or more other nodes on the communication path (18) to stop transmitting messages on the communication path (18) for a predetermined time interval.
13. A sensor node for a control system, comprising:
 - circuitry for interfacing a sensor (20) to a communication path (18) of the control system (10);
 - circuitry for obtaining a sensor data sample from the sensor (20) and substantially contemporaneously generating a sensor time-stamp such that the sensor data sample and the sensor time-stamp enable an accurate computation of an actuator control value to be applied to an actuator (22) in the control system (10).
14. The sensor node of claim 13, wherein the sensor data sample and the sensor time-stamp enable an accurate computation of an actuator time-stamp that indicates a time at which the actuator control value is to be applied to the actuator (22).
15. The sensor node of claim 13, wherein the circuitry for obtaining a sensor data sample from the sensor (20) and substantially contemporaneously generating a sensor time-stamp includes a clock circuit (30).
16. The sensor node of claim 13, wherein the sensor node (14) obtains the sensor data sample and generates the sensor time-stamp in response to a request message received over the communication path (18).
17. The sensor node of claim 13, wherein the sensor node (14) periodically obtains the sensor data sample and generates the sensor time-stamp and then transfers the sensor data sample and the sensor time-stamp over the communication path (18).
18. An actuator node for a control system, comprising:
 - circuitry for interfacing an actuator (22) to a communication path (18) of the control system (10);
 - circuitry for triggering the application of an actuator control value to the actuator (22) using an actuator time-stamp.
19. The actuator node of claim 18, further comprising means for determining the actuator control value and the actuator time-stamp in response to a sensor data sample and a sensor time-stamp received over the communication path (18).
20. The actuator node of claim 18, wherein the circuitry for triggering the application of an actuator control value includes a clock circuit (32).

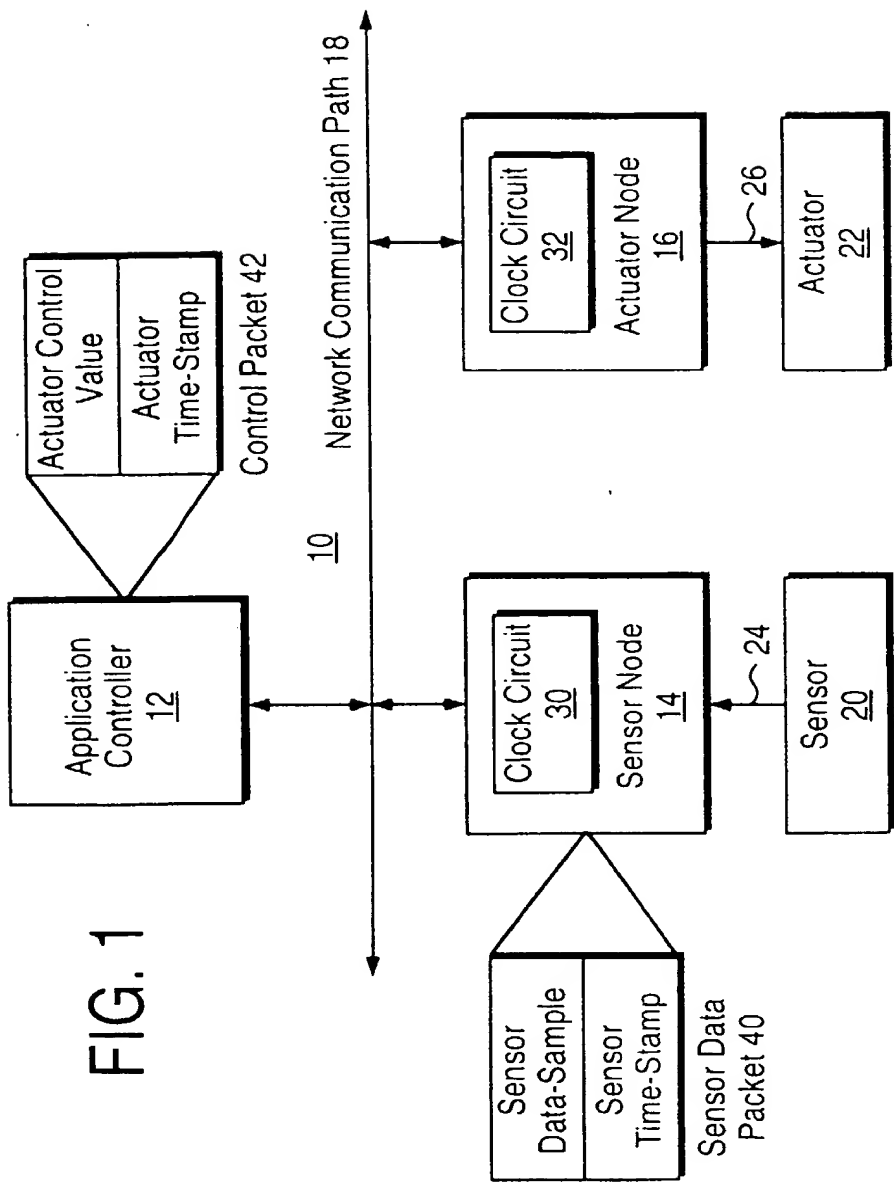


FIG. 2

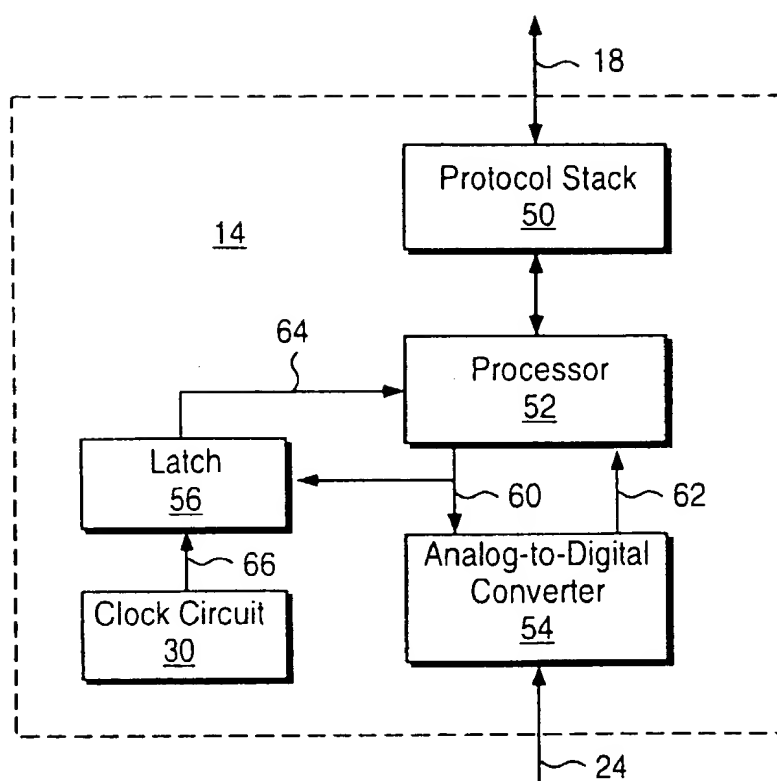


FIG. 3

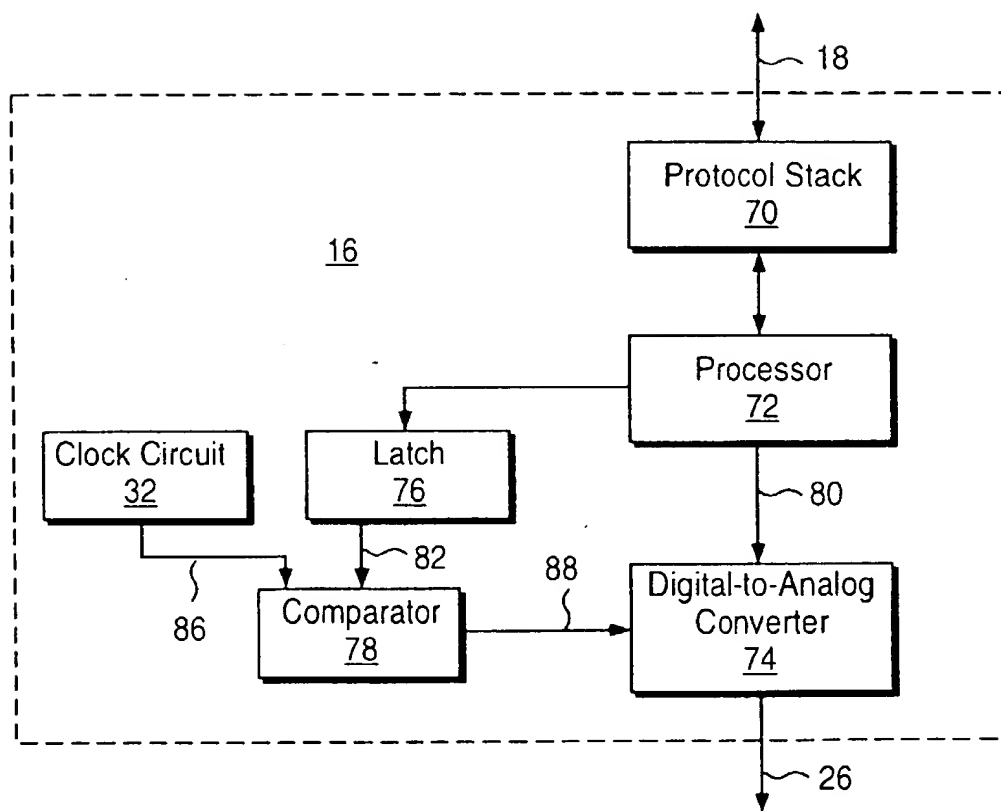


FIG. 4

